ICS 233 – Fall 2010

Computer Architecture and Assembly Language

Programming Assignment 3

Gaussian elimination is a well-known technique for solving simultaneous linear systems of equations. Variables are eliminated one by one until there is only one left, and the discovered values of variables are back-substituted to obtain the values of other variables. In practice, the linear equations are represented as an augmented matrix *A*[*N*][*N*+1] with *N* rows and *N*+1 columns. The matrix is converted to an upper triangular matrix. Then back substitutions are used to produce the solution vector. Pseudo-code for *Gaussian* elimination is shown below.

```
procedure Gaussian (int N, float A[N][N+1]) {
   for k = 0 to n-2 do {
      for i = k+1 to N-1 do {
         factor = A[i][k]/A[k][k];
         for j = k+1 to N do {
            A[i][j] = A[i][j] – factor * A[k][j];
         }
         A[i][k] = 0
      }
   }
}
```

Pseudo-code for the *solve* procedure is shown below. This procedure is called after Gaussian elimination. It receives as input the upper triangular converted matrix *A*, and produces as output the solution vector *Sol*.

```
procedure Solve(int N, float A[N][N+1], float Sol[N]) {
  for (i=N-1; i>=0; i--) {
    Sol[i] = A[i][N];
    for (j=i+1; j<N; j++) {
      Sol[i] = Sol[i] - A[i][j] * Sol[j];
    }
    Sol[i] = Sol[i]/A[i][i];
  }
}
```

Write a MIPS assembly language program to perform Gaussian Elimination of floating-point matrices of size *N* by *N*+1, and to produce a solution vector of *N* floating-point elements (Maximum value of *N* is 10). Ask the user to input the matrix values. One large array *A* should be defined in the data segment, but you can operate on a subset of the array elements. Perform Gaussian elimination on *A* and produce a solution vector. *All arithmetic operations should be done using the floating-point instructions*. Display the solution vector, where each number is displayed on a separate line. A sample run is show below:

```
Enter Matrix Size N (range 2 to 10): 3
Matrix Input (3 rows by 4 values):
. . .
Solution Vector (5 values):
. . .
```